



# BREVET D'INVENTION

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

## COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le 05 JUL 2001

Pour le Directeur général de l'Institut  
national de la propriété industrielle  
Le Chef du Département des brevets

Martine PLANCHE

INSTITUT  
NATIONAL DE  
LA PROPRIÉTÉ  
INDUSTRIELLE

SIEGE  
26 bis, rue de Saint Petersburg  
75800 PARIS cedex 08  
Téléphone : 01 53 04 53 04  
Télécopie : 01 42 93 59 30  
<http://www.inpi.fr>

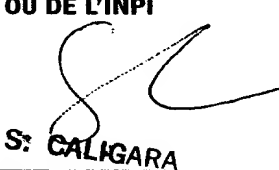
**This Page Blank (uspto)**

REQUÊTE EN DÉLIVRANCE 1/2

Cet imprimé est à remplir lisiblement à l'encre noire

DB 540 W / 260899

<p><b>REMISE DES PIÈCES</b> DATE LIEU <b>31 MAI 2000</b> N° D'ENREGISTREMENT NATIONAL ATTRIBUÉ PAR L'INPI DATE DE DÉPÔT ATTRIBUÉE PAR L'INPI</p> <p><b>0006976</b> <b>31 MAI 2000</b></p>		<p><b>1 NOM ET ADRESSE DU DEMANDEUR OU DU MANDATAIRE À QUI LA CORRESPONDANCE DOIT ÊTRE ADRESSÉE</b></p> <p>CABINET BALLOT SCHMIT 9 rue Claude Chappe Technopôle Metz 2000 57070 METZ</p>	
<p><b>V s références pour ce dossier</b> (facultatif) 015594</p>			
<p><b>C nfirmation d'un dépôt par télécopie</b> <input type="checkbox"/> N° attribué par l'INPI à la télécopie</p>			
<p><b>2 NATURE DE LA DEMANDE</b></p>		<p><b>Cochez l'une des 4 cases suivantes</b></p>	
Demande de brevet		<input checked="" type="checkbox"/>	
Demande de certificat d'utilité		<input type="checkbox"/>	
Demande divisionnaire		<input type="checkbox"/>	
<p><i>Demande de brevet initiale</i> <i>ou demande de certificat d'utilité initiale</i></p>		<p>N° _____ Date ____/____/____ N° _____ Date ____/____/____</p>	
Transformation d'une demande de brevet européen <i>Demande de brevet initiale</i>		<p><input type="checkbox"/> N° _____ Date ____/____/____</p>	
<p><b>3 TITRE DE L'INVENTION (200 caractères ou espaces maximum)</b></p> <p>Procédé de préparation et d'exécution d'une procédure d'autotest et procédé de génération d'autotest associé.</p>			
<p><b>4 DÉCLARATION DE PRIORITÉ OU REQUÊTE DU BÉNÉFICE DE LA DATE DE DÉPÔT D'UNE DEMANDE ANTÉRIEURE FRANÇAISE</b></p>		<p>Pays ou organisation _____ N° _____ Date ____/____/____ Pays ou organisation _____ N° _____ Date ____/____/____ Pays ou organisation _____ N° _____ Date ____/____/____ <input type="checkbox"/> S'il y a d'autres priorités, cochez la case et utilisez l'imprimé «Suit »</p>	
<p><b>5 DEMANDEUR</b></p>		<p><input type="checkbox"/> S'il y a d'autres demandeurs, cochez la case et utilisez l'imprimé «Suite»</p>	
Nom ou dénomination sociale		STMICROELECTRONICS S.A.	
Prénoms			
Forme juridique		SA	
N° SIREN		. . . . .	
Code APE-NAF		. . . . .	
Adresse	Rue	7 Avenue Galliéni	
	Code postal et ville	94250	GENTILLY
Pays		France	
Nationalité		FR	
N° de téléphone (facultatif)			
N° de télécopie (facultatif)			
Adresse électronique (facultatif)			

REMISE DES PIÈCES DATE LIEU <b>31 MAI 2000</b> <b>54 INPI NANCY</b> N° D'ENREGISTREMENT NATIONAL ATTRIBUÉ PAR L'INPI		Réservé à l'INPI		0006976		DB 540 W / 260899	
<b>Vos référ nces pour ce dossier :</b> <i>(facultatif)</i>				015594			
<b>6 MANDATAIRE</b>							
Nom				LECLAIRE			
Prénom				Jean-Louis			
Cabinet ou Société				CABINET BALLOT SCHMIT			
N °de pouvoir permanent et/ou de lien contractuel							
Adresse		Rue		9 rue Claude Chappe Technopôle Metz 2000			
		Code postal et ville		57070		METZ	
N° de téléphone <i>(facultatif)</i>				03 87 74 81 36			
N° de télécopie <i>(facultatif)</i>				03 87 36 26 76			
Adresse électronique <i>(facultatif)</i>							
<b>7 INVENTEUR (S)</b>							
Les inventeurs sont les demandeurs				<input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non <b>Dans ce cas fournir une désignation d'inventeur(s) séparée</b>			
<b>8 RAPPORT DE RECHERCHE</b>				<b>Uniquement pour une demande de brevet (y compris division et transformati n)</b>			
Établissement immédiat ou établissement différé				<input checked="" type="checkbox"/> <input type="checkbox"/>			
Paiement échelonné de la redevance				<b>Paiement en trois versements, uniquement pour les personnes physiques</b> <input type="checkbox"/> Oui <input type="checkbox"/> Non			
<b>9 RÉDUCTION DU TAUX DES REDEVANCES</b>				<b>Uniquement pour les personnes physiques</b> <input type="checkbox"/> Requête pour la première fois pour cette invention <i>(joindre un avis de non-imposition)</i> <input type="checkbox"/> Requête antérieurement à ce dépôt <i>(joindre une copie de la décision d'admission pour cette invention ou indiquer sa référence) :</i>			
Si vous avez utilisé l'imprimé «Suite», indiquez le nombre de pages jointes							
<b>10 SIGNATURE DU DEMANDEUR OU DU MANDATAIRE</b> (Nom et qualité du signataire) LECLAIRE Jean-Louis 93.4009				<b>VISA DE LA PRÉFECTURE OU DE L'INPI</b>  S: CALIGARA			

DÉPARTEMENT DES BREVETS

26 bis, rue de Saint Pétersbourg

75800 Paris Cedex 08

Téléphone : 01 53 04 53 04 Télécopie : 01 42 93 59 30

DÉSIGNATION D'INVENTEUR(S) Page N° 1. / 1.

(Si le demandeur n'est pas l'inventeur ou l'unique inventeur)

Cet imprimé est à remplir lisiblement à l'encre noire

DB 113 W / 260899

<b>Vos références pour ce dossier</b> (facultatif)		015594	
<b>N° D'ENREGISTREMENT NATIONAL</b>		0006976	
<b>TITRE DE L'INVENTION</b> (200 caractères ou espaces maximum)			
Procédé de préparation et d'exécution d'une procédure d'autotest et procédé de génération d'autotest associé.			
<b>LE(S) DEMANDEUR(S) :</b>			
STMICROELECTRONICS S.A. 7 Avenue Galliéni 94250 GENTILLY France			
<b>DESIGNE(NT) EN TANT QU'INVENTEUR(S) :</b> (Indiquez en haut à droite «Page N° 1/1» S'il y a plus de trois inventeurs, utilisez un formulaire identique et numérotez chaque page en indiquant le nombre total de pages).			
<b>Nom</b>		MATHIEU	
<b>Prénoms</b>		Frédéric	
<b>Adresse</b>	<b>Rue</b>	CABINET BALLOT SCHMIT 9 rue Claude Chappe	
	<b>Code postal et ville</b>	57070	METZ
<b>Société d'appartenance</b> (facultatif)			
<b>Nom</b>		BARD	
<b>Prénoms</b>		Frédéric	
<b>Adresse</b>	<b>Rue</b>	CABINET BALLOT SCHMIT 9 rue Claude Chappe	
	<b>Code postal et ville</b>	57070	METZ
<b>Société d'appartenance</b> (facultatif)			
<b>Nom</b>			
<b>Prénoms</b>			
<b>Adresse</b>	<b>Rue</b>		
	<b>Code postal et ville</b>		
<b>Société d'appartenance</b> (facultatif)			
<b>DATE ET SIGNATURE(S)</b> <b>DU (DES) DEMANDEUR(S)</b> <b>OU DU MANDATAIRE</b> (Nom et qualité du signataire) LECLAIRE Jean-Louis 93.4009		<b>CABINET BALLOT-SCHMIT</b> <b>CONSEILS EN PROPRIÉTÉ INDUSTRIELLE</b> 9, rue Claude Chappe Technopôle Metz 2000 57070 METZ	

**PROCEDE DE PREPARATION ET D'EXECUTION D'UNE PROCEDURE**  
**D'AUTOTEST ET PROCEDE DE GENERATION D'AUTOTESTS ASSOCIE**

L'invention a pour objet un procédé de préparation et d'exécution d'une procédure d'autotest et un procédé de génération d'autotests associé, pour la vérification fonctionnelle et de performance de circuits logiques, notamment de modèles de processeur.

Avec le développement croissant des systèmes électroniques embarqués utilisant des processeurs programmables, il y a une forte demande d'outils fiables, précis et réutilisables pour développer ces processeurs. Les outils demandés sont par exemple des compilateurs optimisateurs ou des modèles (également appelés simulateurs) capables de décrire bit par bit ou cycle par cycle le fonctionnement du processeur pour permettre l'exploration de son architecture et sa validation.

Le développement et la mise au point d'un processeur sur circuit intégré nécessitent la réalisation de simulateurs. Un simulateur est un programme qui permet de simuler tout ou partie du comportement du processeur physique qui en découlera. Pour le développement d'un même processeur par exemple, des simulateurs de fonctionnalité sont utilisés pour contrôler le jeu d'instructions du processeur, et des simulateurs de performances sont utilisés pour étudier le fonctionnement global du processeur physique à réaliser. Selon la complexité (en terme de fonctionnalités ou de performance) du processeur physique qui doit être réalisé, un ou plusieurs simulateurs sont développés. Lorsque les simulateurs sont au point, leur dernière version est transformée en un circuit intégré qui correspond au processeur physique. Les simulateurs sont en général développés par des équipes différentes et dans des langages de programmation différents, le plus souvent pour des raisons de facilité de développement. Dans ce

qui suit, l'expression "modèle de processeur" sera utilisée pour désigner de manière générique un processeur et les simulateurs créés pour le développer.

Pour assurer la fiabilité du processeur final, il est indispensable, à chaque étape du développement, de valider le modèle de processeur obtenu à chaque étape, c'est-à-dire de vérifier que le modèle de processeur (simulateur ou circuit intégré final) obtenu à chaque étape, a bien les caractéristiques et le comportement souhaités. La validation se fait par l'intermédiaire de procédures de test spécifiques. Par exemple, au cours d'une procédure de test de fonctionnalité, il est demandé au modèle de processeur d'exécuter une unique instruction de son jeu d'instructions pour contrôler son fonctionnement correct. De la même façon, au cours de l'exécution d'une procédure de test de performance, plusieurs instructions sont exécutées successivement pour contrôler l'enchaînement correct des instructions. Généralement, plusieurs procédures de test sont effectuées successivement et de préférence de manière automatique. Une procédure de test est généralement mise en œuvre sous la forme d'un programme (appelé "test") exécutable par le modèle de processeur à tester.

Après exécution d'un test, un résultat est obtenu qui correspond par exemple au contenu des éléments du modèle de processeur après exécution du test, ou bien à l'évolution du contenu des éléments du modèle à chaque cycle d'horloge et pendant toute la durée d'exécution du test. Par éléments du modèle de processeur, il faut comprendre ici des éléments tels que les mémoires ou les registres du modèle à tester. Le résultat du test est en général mémorisé dans un fichier de résultat, appelé trace de la procédure de test. Le contenu du fichier de résultat (type d'informations, ordre, ...) est défini dans la procédure de test.

Vu la complexité des modèles de processeur actuels, et donc le nombre de tests à réaliser pour les valider correctement et de manière fiable, il n'est plus possible d'écrire les procédures de test ou les tests à la main.

5 Des procédés de génération de tests sont utilisés, qui fournissent des procédures de test, ou des tests, en fonction de paramètres imposés par l'utilisateur. Les paramètres sont par exemple le nom et la définition fonctionnelle d'une instruction du jeu d'instructions du  
10 modèle de processeur à tester, ou bien l'ordre dans lequel deux instructions successives doivent être exécutées. De tels procédés de génération de tests sont mis en œuvre par des générateurs de tests, qui sont des programmes réalisant les différentes étapes des procédés  
15 associés.

Un générateur de tests est en général développé pour un modèle de processeur particulier, et il fournit des procédures de test dans un langage compréhensible d'une part par le modèle à tester, et d'autre part par  
20 l'interface de contrôle du modèle et les outils utilisés pour le développer. Actuellement pour le développement d'un même processeur, au moins un générateur de tests est développé pour chaque type de modèle de processeur, par exemple un pour le simulateur de performance, un pour le  
25 processeur final, etc. Bien que, souvent, les tests à réaliser sont identiques sur le fond, deux générateurs de tests vont fournir des procédures de test différentes, adaptées aux modèles de processeur auxquelles elles sont destinées. Pendant toute la durée du développement du  
30 processeur, les procédures de test fournies par les différents générateurs de tests sont mémorisées, ce qui nécessite des capacités de stockage de données importantes.

De plus, la trace (le résultat) fournie par une  
35 procédure de test est dépendante de la manière dont la procédure a été obtenue. En particulier, deux procédures



de test obtenues à partir de deux générateurs de tests différents, vont peut-être donner des traces différentes, en terme de type d'information ou de format notamment.

Par exemple, pour tester la réalisation d'une  
5 addition  $C=A+B$ , une première procédure de test fournit une trace indiquant le contenu du registre C à la fin de l'exécution du test, alors qu'une deuxième procédure fournit une trace indiquant l'évolution du contenu du registre C, tout au long de la réalisation du même test.  
10 Dans de tels cas, il n'est pas possible de comparer directement les deux traces, et il est nécessaire de passer par l'intermédiaire d'un traducteur ou d'un interpréteur.

De plus, la trace d'une procédure de test ne peut  
15 pas être exploitée directement par un générateur de tests actuel, car il ne connaît pas le résultat attendu de la procédure de test qu'il a généré et il ne sait pas l'obtenir. En conséquence, il est nécessaire de trier et valider toutes les procédures de test fournies par un  
20 générateur, afin de ne garder que les tests corrects, voire même uniquement les tests dont le résultat est intéressant pour l'utilisateur. En effet, des tests fournis par un générateur de tests peuvent être redondants, d'autres apportent très peu d'information, ou  
25 bien des erreurs de programmation apparaissent dans certains tests.

Pour valider et utiliser les tests fournis par les générateurs de tests actuels, il est nécessaire de connaître le résultat attendu des procédures de test  
30 qu'ils exécutent. Pour cela, on utilise généralement un ou des modèles de processeur de référence, parfaits, qui fournissent le résultat attendu pour chaque test ou procédure de test. Pour vérifier si le modèle de processeur à tester est correct ou pas, il faut alors  
35 comparer la trace fournie par une procédure de test exécutée par le modèle, avec la trace fournie par la même

procédure de test lorsqu'elle est exécutée par un modèle de référence.

Des difficultés peuvent alors apparaître pour effectuer la comparaison et il peut être nécessaire  
5 d'avoir recours à un interpréteur ou un traducteur pour pouvoir effectuer cette comparaison. De plus, la précision des générateurs de tests actuels dépend de la précision des simulateurs de référence disponibles. Or, il est difficile d'obtenir des modèles de référence  
10 parfaits, qui doivent bien sûr être eux-mêmes validés, et il est encore plus difficile d'obtenir un modèle de référence qui soit capable d'utiliser toutes les instructions, toutes les fonctionnalités d'un modèle de processeur à tester.

15

Au vu des inconvénients des générateurs de tests actuels, un but de l'invention est de mettre en œuvre un procédé qui permet d'obtenir rapidement des procédures de test et des tests utilisables pour les différents types  
20 de modèles (simulateurs ou processeur final) nécessaires au développement d'un même processeur, sans adaptation ni modification. Ceci limite alors le nombre de tests nécessaires au développement complet d'un processeur, c'est-à-dire à la validation de tous les modèles de  
25 processeur utilisés pour son développement, y compris le processeur final. Le temps nécessaire à la validation de tous les modèles ainsi que les besoins de stockage des différents tests sont alors également réduits.

Un autre but de l'invention est de mettre en œuvre  
30 un procédé qui permet à l'utilisateur d'obtenir des procédures de test et des tests qui fournissent des résultats sous une forme simple et directement exploitable avec tous les outils de développement d'un même processeur.

35 Avec ces objectifs en vue, l'invention a pour objet un procédé de préparation et d'exécution d'une procédure

d'autotest pour valider le comportement d'un modèle de processeur à tester, le modèle de processeur étant un processeur ou un simulateur associé, le procédé étant caractérisé en ce qu'il comprend les étapes suivantes, consistant à :

- recevoir des spécifications (E2) d'un utilisateur concernant au moins une instruction à tester d'un jeu d'instructions du modèle de processeur,

- lire (E4), dans une table, des données caractéristiques du modèle de processeur à tester, les données comprenant notamment une définition fonctionnelle de l'instruction à tester et une définition fonctionnelle des éléments du modèle de processeur,

- calculer un résultat attendu (E6) à la fin de l'exécution de l'instruction à tester, le calcul étant réalisé à partir des spécifications de l'utilisateur et des données caractéristiques du modèle de processeur, et

- faire réaliser, par le modèle à tester, une procédure d'autotest (E8) pour valider l'instruction à tester, la procédure d'autotest fournissant en retour un mot de résultat qui est égal à une première valeur (OK) si le comportement du modèle de processeur est correct, et qui est égal à une deuxième valeur (ERROR) sinon.

Ainsi, avec l'invention, le résultat est directement exploitable par l'utilisateur, sans nécessiter d'étape de comparaison ou de traduction complémentaire par exemple. Un résultat égal à OK signifie en général d'une part que la procédure d'autotest E8 est correctement mise en œuvre, et d'autre part que le modèle de processeur sous test réagit correctement par rapport à la ou les instructions à exécuter.

Inversement, un résultat égal à ERROR signifie soit que le modèle de processeur n'est pas correct par rapport aux spécifications attendues, soit que la procédure d'autotest n'est pas correctement mise en œuvre (choix

peu judicieux des instructions à tester, erreur d'écriture pour la mise en œuvre de la procédure, paramètres incorrects, etc.).

De préférence, la procédure d'autotest (E8) comprend les sous-étapes suivantes, consistant à:

- initialiser (E81) des éléments du modèle de processeur à tester,
- exécuter l'instruction à tester (E82), et obtenir un résultat,
- comparer le résultat obtenu (E83) et le résultat attendu, et
- fournir (E84) un mot de résultat de la comparaison (OK/ERROR).

Selon une variante, une seule instruction du jeu d'instructions du modèle de processeur est exécutée, ce qui permet de tester le comportement fonctionnel du modèle. Selon une autre variante, au moins deux instructions du jeu d'instructions du modèle de processeur sont exécutées successivement, pour valider les performances du modèle de processeur à tester.

Ainsi, une procédure d'autotest est une procédure de test englobant une initialisation des éléments (registres et/ou mémoires) du modèle de processeur à tester, une ou plusieurs instructions à exécuter, un calcul d'un résultat attendu et une comparaison du résultat attendu avec un résultat obtenu après exécution de la ou des instructions à exécuter, la procédure de test fournissant un résultat simple à interpréter, de type vrai/faux.

Une procédure d'autotest est donc une procédure de test qui est autonome car elle est exécutée sans autre instruction qu'une instruction de début et elle n'a pas besoin d'être validée avant d'être utilisée. Avec l'invention, il n'est donc plus nécessaire de disposer de modèle de référence pour valider les procédures d'autotests.

Selon un autre mode de réalisation de l'invention, le procédé comprend les étapes suivantes, consistant à :

- faire réaliser, par le modèle à tester, une procédure d'autotest (E8) et recevoir un mot de résultat,
- 5       - prendre une décision (E10) : si le résultat précédent est égal à la première valeur alors réaliser une étape E12, sinon, si le résultat précédent est égal à la deuxième valeur (ERROR), réaliser une étape E14,
- 10       - prendre une décision (E12) : si une autre procédure d'autotest doit être exécutée, alors réaliser une nouvelle étape E8, sinon, fin du procédé, et
- 15       - mémoriser (E14) des informations relatives à la procédure d'autotest précédemment effectuée, les informations contenant notamment une adresse à laquelle une erreur a été détectée, puis fin du procédé.

De préférence, les étapes E8 à E14 sont précédées des étapes d'initialisation suivantes :

- recevoir des spécifications d'un utilisateur (E'2) concernant au moins deux procédures d'autotest,
- 20       - lire (E'4), dans une table, des données caractéristiques du modèle de processeur à tester nécessaires à l'exécution des procédures d'autotest,
- 25       - exécuter (E'6) plusieurs étapes E6, de manière successive, chaque étape (E6) correspondant à une procédure d'autotest.

Ce mode de réalisation présente l'avantage d'exécuter plusieurs procédures d'autotests successives, ce qui limite l'intervention de l'utilisateur, s'il le souhaite, au cas où une procédure d'autotests donne un

30       résultat incorrect.

De préférence, après l'étape E'6, une étape E7 est réalisée, qui fournit à l'utilisateur une étude statistique des instructions à tester lors des étapes E8 suivantes. L'utilisateur sait ainsi précisément quelles

35       instructions vont être testées, combien de fois chaque instruction ou combinaison d'instructions va être

exécutée, en d'autres termes, il connaît précisément, pour l'ensemble des procédures d'autotests qui vont être exécutées, la couverture du jeu d'instructions et des performances du modèle de processeur à valider.

5 De la même façon, une étape E16 peut également être réalisée à la fin du procédé, qui fournit une étude statistique des résultats fournis lors de toutes les procédures d'autotests exécutées par le modèle sous test.

L'invention a également pour objet un procédé de  
10 génération de programmes d'autotest, caractérisé en ce qu'il comprend les étapes E2 à E6 du procédé précédemment décrit, et en ce qu'il comprend en outre l'étape E'8 suivante :

- écrire un programme d'autotest pour faire  
15 exécuter une procédure d'autotest par le modèle de processeur.

Ce programme, mémorisé, peut être utilisé ultérieurement par le modèle de processeur à tester. Pour cela, le programme d'autotest est de préférence écrit  
20 dans un langage de type assembleur, compréhensible par tous les modèles à tester d'un même processeur et les étapes d'initialisation sont réalisées à partir d'instructions du jeu d'instructions du modèle à tester, tous les modèles d'un même processeur ayant le même jeu  
25 d'instructions. Une même procédure d'autotests selon l'invention est ainsi utilisable par tous les modèles d'un même processeur ; le nombre de procédures d'autotests à développer est donc limité de fait puisqu'il n'est pas nécessaire de développer des  
30 procédures spécifiques pour chaque modèle. Le temps de réalisation des procédures d'autotests est donc limité, ainsi que les besoins de stockages de ces procédures.

De plus, une même procédure d'autotests pouvant être effectuée par des modèles de processeurs différents,  
35 l'utilisateur n'a plus besoin de passer par l'intermédiaire d'un interpréteur ou d'un traducteur pour

comparer deux résultats d'un même autotest, réalisés sur deux modèles de processeur différents par exemple.

De préférence enfin, le procédé de génération d'autotests de l'invention est mis en œuvre sous la forme  
5 d'un programme écrit dans un langage évolué de type DGL et/ou C++, compréhensible par l'utilisateur, ce qui facilite le travail de développement et de mise en œuvre de ce procédé.

L'utilisateur peut alors, par des modifications  
10 simples, obtenir facilement des procédures d'autotests et des autotests pour tout type de modèles de processeur, pour le développement de nouveaux processeurs par exemple.

15 L'invention sera mieux comprise et d'autres caractéristiques et avantages apparaîtront à la lecture de la description qui va suivre, la description faisant référence aux dessins annexés dans lesquels :

- la figure 1 est un diagramme d'un algorithme de  
20 mise en œuvre d'un procédé de préparation et d'exécution d'une procédure d'autotest, selon l'invention,

- la figure 2 est un diagramme détaillant une étape de l'algorithme de la figure 1.

- la figure 3 montre des améliorations possibles de  
25 l'algorithme de la figure 1.

Le procédé de la figure 1, selon l'invention, est utilisé pour préparer et exécuter une procédure d'autotest, dans le but de tester le comportement (en  
30 terme de fonctionnalité ou de performance) d'un processeur à valider ou d'un simulateur associé, un simulateur étant un modèle qui permet de simuler le comportement du processeur.

Le procédé est par exemple mis en œuvre par une  
35 interface de commande, de type unité centrale ou station de travail, qui pilote le fonctionnement et les

opérations de tests du modèle à tester qui lui relié, sous la forme d'un programme à tester (simulateur) ou d'un circuit (processeur physique) à tester.

Comme précédemment, on utilisera par la suite  
5 l'expression "modèle de processeur" pour désigner de manière générale un processeur ou un simulateur correspondant.

Le procédé de la figure 1 comprend une étape E2 de réception de spécifications, une étape E4 de lecture de  
10 données, une étape E6 de calcul d'un résultat attendu et une étape E8 de réalisation d'une procédure d'autotest, cette dernière étape étant exécutée par le modèle sous test.

Au cours de la réalisation de l'étape E2,  
15 l'utilisateur précise quel type de test il souhaite effectuer, et en particulier quelle(s) instruction(s) doivent être exécutées par le modèle de processeur à tester. La ou les instructions à tester sont bien sûr des instructions du jeu d'instructions du modèle sous test,  
20 ces instructions étant à priori les seules qu'il peut exécuter. L'utilisateur choisit par exemple une unique instruction s'il souhaite réaliser un test de fonctionnalité du jeu d'instructions du modèle à tester, c'est-à-dire s'il souhaite vérifier que l'instruction  
25 choisie est exécutée correctement par le modèle à tester. L'utilisateur peut également choisir deux instructions ou plus, s'il souhaite effectuer un test de performance du modèle de processeur, en précisant dans quel ordre doivent être exécutées les instructions choisies.

30 Au cours de l'étape E2, l'utilisateur précise également les valeurs initiales à prendre en compte pour exécuter les instructions précédemment choisies. Par exemple, si l'utilisateur souhaite voir exécuter l'addition  $C=A+B$ , alors il va préciser les valeurs  
35 numériques de A et B et les registres dans lesquels elles doivent être mémorisées, le nom de l'instruction à



utiliser ADD, au moins une adresse pour indiquer où doit être mémorisé le résultat, s'il est mémorisé, et éventuellement une adresse où doit être mémorisé un éventuel débordement de capacité.

5        Au cours de l'étape E2, l'utilisateur définit enfin  
quelles informations il souhaite obtenir en retour après  
exécution de la ou des instructions. Dans le cas d'une  
addition, l'utilisateur peut par exemple préciser qu'il  
souhaite obtenir le contenu des registres de résultat et  
10    savoir si un éventuel débordement de capacité a eu lieu.  
Il peut également choisir de demander une information sur  
l'évolution du contenu de tous les registres du modèle à  
tester, cycle par cycle, au cours de l'exécution de  
l'instruction choisie.

15        Au cours de l'étape E4, des paramètres,  
correspondant à des caractéristiques techniques du modèle  
de processeur à tester, sont lus dans une table de  
caractéristiques. Les paramètres lus comprennent  
notamment une description fonctionnelle et  
20    comportementale de la ou des instructions à exécuter, une  
liste des éléments du modèle de processeur à utiliser  
pour exécuter la ou les instructions, et également une  
description fonctionnelle de ces éléments, incluant leur  
adresse, leur taille, etc. Les paramètres lus dépendent  
25    bien sûr des spécifications de l'utilisateur précisées  
dans l'étape E2.

La table de caractéristiques contient toutes les  
caractéristiques techniques du modèle de processeur sous  
test. La table comprend notamment une description  
30    fonctionnelle et comportementale précise de tous les  
éléments du modèle (nombre de registres, de mémoire,  
d'unités de calcul, etc., taille, adresses de tous les  
éléments, connexions les uns aux autres, etc.). La table  
comprend également une description complète du jeu  
35    d'instructions du modèle et des règles de compilation  
associées.

Toutes les informations contenues dans la table peuvent être fournies par l'utilisateur, lors d'une étape E0 (non représentée sur la figure 1) d'initialisation du procédé par exemple. Toutes ces informations peuvent également être obtenues et mémorisées par ailleurs, dans une mémoire.

Au cours de l'étape E6, un résultat théorique est calculé, qui correspond au résultat à attendre si le test est effectué correctement et si le modèle à tester est correct. Le résultat théorique calculé dépend, d'une part des spécifications de l'utilisateur précisées au cours de l'étape E2, et d'autre part des caractéristiques techniques du modèle sous test lues au cours de l'étape E4. Le résultat attendu est ainsi calculé directement à partir des caractéristiques du processeur. Il n'est donc pas nécessaire de disposer d'un modèle de processeur de référence pour obtenir ce résultat attendu, ce qui est particulièrement avantageux.

Dans le cas de l'addition  $C=A+B$  décrit ci-dessus, le résultat théorique calculé comprend le nombre C sous forme binaire qui doit être contenu dans les registres de résultat si le test est bon, et le contenu d'un registre indiquant si un dépassement de capacité doit avoir lieu.

Les étapes E2, E4 et E6 ci-dessus préparent la réalisation de la procédure d'autotest E8 suivante, qui est exécutée par le modèle à tester, selon les instructions de l'interface de commande qui met en œuvre le procédé de l'invention.

La procédure d'autotest E8 comprend, conformément à la figure 2, une étape d'initialisation E81, une étape d'exécution d'instructions E82, une étape d'étape de comparaison E83 et une étape de fourniture de résultat E84.

Au cours de l'étape d'initialisation E81, tous les éléments (registres, mémoires, circuits de calcul, etc.) du modèle de processeur sous test sont initialisés. Par

exemple, les valeurs initiales, si elles existent, sont chargées dans les registres correspondants, les registres ne recevant pas de valeur initiale reçoivent un zéro, les circuits de calcul sont initialisés, etc.

5        Au cours de l'étape d'exécution d'instructions E82, la ou les instructions à tester sont exécutées, conformément aux spécifications de l'utilisateur précisées dans l'étape E2, et un résultat est extrait du modèle sous test. Les informations extraites sont celles  
10        précisées par l'utilisateur au cours de l'étape E2. Par exemple, dans le cas de l'opération  $C=A+B$ , le résultat obtenu comprend le nombre C, contenu dans un ou plusieurs registres de résultat, et une information indiquant si un dépassement de capacité a eu lieu ou pas lors de la  
15        réalisation de l'étape E82.

      Lors de la réalisation de l'étape E83 suivante, une comparaison est effectuée, entre le résultat calculé au cours de l'étape E6 et le résultat réellement obtenu au cours de l'étape E82.

20        Un résultat de la comparaison est finalement fourni (étape E84), et éventuellement mémorisé. Le résultat de la comparaison prend deux valeurs, qui sont par exemple OK si la comparaison indique que les résultats attendu et obtenu sont les mêmes, et ERROR dans le cas contraire.

25        Un résultat égal à OK signifie en général d'une part que la procédure d'autotest E8 est correctement mise en œuvre, et d'autre part que le modèle de processeur sous test réagit correctement par rapport à la ou les instructions à exécuter.

30        Inversement, un résultat égal à ERROR signifie soit que le modèle de processeur n'est pas correct par rapport aux spécifications attendues, soit que la procédure d'autotest n'est pas correctement mise en œuvre (choix peu judicieux des instructions à tester, erreur  
35        d'écriture pour la mise en œuvre de la procédure, paramètres incorrects, etc.).

Le diagramme de la figure 1 décrit ci-dessus montre une réalisation simple d'un procédé de préparation et d'exécution d'une procédure d'autotest, selon  
 5 l'invention. Dans un mode préféré de réalisation, une procédure d'autotest E8 est mise en œuvre sous la forme d'un programme d'autotest (plus simplement appelé autotest).

L'exemple simple de la figure 1 peut être amélioré  
 10 de différentes manières, sans sortir du cadre de l'invention. Des améliorations possibles sont représentées sur la figure 3.

Une première amélioration permet d'effectuer  
 15 plusieurs procédures d'autotest E8 successivement, de manière automatique, ce qui permet un gain de temps considérable.

Au cours de l'étape E'2, l'utilisateur précise, comme précédemment, l'ensemble des instructions à tester, l'ordre dans lequel elles doivent être testées, les  
 20 valeurs initiales à prendre en compte et les résultats à extraire, ceci pour chaque procédure d'autotest à effectuer.

Au cours de l'étape E'4, tous les paramètres nécessaires à la réalisation de toutes les procédures  
 25 d'autotest sont lus dans la table.

Au cours de l'étape E'6, sont calculés de manière successive, les résultats attendus pour chaque procédure d'autotest.

Les étapes E'2, E'4 et E'6 décrites ci-dessus  
 30 préparent la réalisation des étapes E8, E10, E12 et E14 suivantes.

Une première étape E8 est réalisée, identique à celle décrite dans l'exemple simple de la figure 1. Une première étape de décision E10 est ensuite réalisée.

35 Si le résultat de l'étape E84 précédente est correct (OK), alors une deuxième étape de décision E12

est effectuée, pour vérifier si une nouvelle procédure d'autotest E8 doit être effectuée ou pas.

Si le résultat de l'étape E12 est positif, alors une nouvelle étape E8 est effectuée. Dans le cas  
5 contraire, le procédé se termine.

Si le résultat de l'étape E84 précédente est mauvais (ERROR), alors une étape E14 de mémorisation est effectuée, au cours de laquelle des informations relatives à la procédure d'autotest incorrecte sont mémorisées. Les  
10 informations comprennent notamment l'adresse où l'erreur a été détectée. Ces informations sont obtenues à partir de l'état architectural du processeur sous test.

Le procédé se termine alors à la fin de l'étape de mémorisation E14. Ainsi le procédé se termine dès qu'une  
15 procédure d'autotest fournit un mauvais résultat. L'utilisateur sait alors immédiatement quelle procédure d'autotest pose problème ou quelles fonctionnalités et / ou performance du modèle sous test ne sont pas conformes à ses souhaits.

20 Dans un mode préféré de réalisation, l'ensemble des étapes E8, E10, E12 et E14 est mis en œuvre sous la forme d'un programme d'autotest, plus simplement appelé autotest.

Selon une variante du procédé de la figure 3,  
25 l'étape de décision E12 est également réalisée après l'étape de mémorisation E14 (ajout de la liaison en pointillés entre l'étape E14 et E12 sur la figure 3, et suppression de la liaison entre les étapes E14 et E16 ).

Dans ce cas, toutes les procédures d'autotest E8 du  
30 procédé sont effectuées, quel que soit le résultat (OK / ERROR) de l'une de ces procédures E8. De plus, chaque fois qu'une procédure E8 donne un mauvais résultat ERROR, alors les informations relatives à cette procédure sont mémorisées (étape E14). Le procédé se termine lorsque  
35 toutes les procédures d'autotest E8 ont été réalisées.

L'ensemble des informations mémorisées au cours de la réalisation des étapes E14 permet à l'utilisateur de voir ultérieurement quelle procédure d'autotest pose problème.

5        Le procédé de la figure 3 est amélioré par l'ajout d'étapes E7 et / ou E16 de fourniture de statistiques (représentées en pointillés sur la figure 3), qui sont notamment intéressantes lorsque plusieurs procédures d'autotest sont effectuées successivement.

10        Au cours de l'étape E7, est effectuée après l'étape E'6 de calculs des résultats attendus, une étude statistique de l'ensemble des instructions qui seront utilisées lors de la réalisation des étapes d'autotest E8  
15        suivantes. Ce résultat permet notamment à l'utilisateur de connaître la couverture des instructions testées, c'est-à-dire de répondre notamment à des questions de type : nombre de fois où une même instruction est utilisée ? toutes les instructions du jeu d'instructions du modèle de processeur sont-elles utilisées ? toutes les  
20        combinaisons de deux (ou plus) instructions sont-elles testées ?, etc.

De manière similaire, l'étape E16 fournit une étude statistique des résultats des procédures d'autotest précédemment effectuées et donne à l'utilisateur une  
25        réponse à des questions de type : nombre de procédures d'autotest fournissant un résultat mauvais ? quelles instructions sont utilisées par les procédures qui fournissent un résultat mauvais ? etc.

De telles études statistiques ne sont pas  
30        indispensable à la mise en œuvre de l'invention, mais elles aident l'utilisateur dans ses choix de procédures d'autotest à effectuer. Elles lui permettent également de déterminer si le modèle de processeur sous test répond aux spécifications attendues ou pas.

Comme il a été dit précédemment, une procédure d'autotest telle que l'étape E8 du procédé de la figure 1 est mis en œuvre sous la forme d'un programme d'autotest, ou autotest.

5        Pour cela l'invention a également pour objet un procédé de génération d'autotest qui comprend les étapes E2, E4, E6 telles que celles décrites en relation avec la figure 1, et qui comprend en outre une étape E'8 d'écriture d'un programme d'autotest pour exécuter une  
10        procédure d'autotest comportant les sous - étapes E81 à E84 précédemment décrites.

Bien sûr, si un programme d'autotest doit être réalisé pour le procédé de la figure 3, il prendra en compte l'ensemble des étapes E8, E10, E12 et E14 à  
15        réaliser.

De préférence, le procédé de génération d'autotest fournit des autotests écrits dans un langage de type assembleur, qui présente l'avantage d'être exécutable par l'ensemble des modèles de processeur à tester, simulateurs utilisés pour le développement ou bien  
20        processeur final par exemple.

Pour un même processeur, deux modèles de processeur ont les mêmes caractéristiques techniques. Aussi, avec l'invention, il n'est alors plus nécessaire d'écrire des autotests différents pour des modèles différents, ce qui permet de réduire considérablement le temps nécessaire pour réaliser ces autotests, ainsi que les capacités de stockage nécessaires pour les mémoriser.

De plus, si un nouveau processeur doit être  
30        développé, alors il suffit de modifier les paramètres de la table de caractéristiques utilisée au cours de l'étape E4, puis d'exécuter une nouvelle fois le procédé de génération d'autotests, pour obtenir simplement et rapidement des autotests pour le nouveau processeur et  
35        ses modèles associés.

## REVENDICATIONS

• 1. Procédé de préparation et d'exécution d'une  
procédure d'autotest pour valider le comportement d'un  
modèle de processeur à tester, le modèle de processeur  
étant un processeur ou un simulateur associé, le procédé  
5 étant caractérisé en ce qu'il comprend les étapes  
suivantes, consistant à :

- recevoir des spécifications (E2) d'un utilisateur  
concernant au moins une instruction à tester d'un jeu  
d'instructions du modèle de processeur,
- 10 - lire (E4), dans une table, des données  
caractéristiques du modèle de processeur à tester, les  
données comprenant notamment une définition fonctionnelle  
de l'instruction à tester et une définition fonctionnelle  
des éléments du modèle de processeur,
- 15 - calculer un résultat attendu (E6) à la fin de  
l'exécution de l'instruction à tester, le calcul étant  
réalisé à partir des spécifications de l'utilisateur et  
des données caractéristiques du modèle de processeur, et
- faire réaliser, par le modèle à tester, une  
20 procédure d'autotest (E8) pour valider l'instruction à  
tester, la procédure d'autotest fournissant en retour un  
mot de résultat qui est égal à une première valeur (OK)  
si le comportement du modèle de processeur est correct,  
et qui est égal à une deuxième valeur (ERROR) sinon.

25

• 2. Procédé selon la revendication 1, caractérisé en  
ce que la procédure d'autotest (E8) comprend les sous-  
étapes suivantes, consistant à :

- initialiser (E81) des éléments du modèle de  
30 processeur à tester,
- exécuter l'instruction à tester (E82), et  
obtenir un résultat,
- comparer le résultat obtenu (E83) et le  
résultat attendu, et



- fournir (E84) un mot de résultat de la comparaison (OK/ERROR).

3. Procédé selon l'une des revendications 1 à 2, caractérisé en ce qu'au moins deux instructions du jeu d'instructions du modèle de processeur sont exécutées successivement, pour valider les performances du modèle de processeur à tester.

4. Procédé selon la revendication 1, caractérisé en ce qu'il comprend les étapes suivantes, consistant à :

- faire réaliser, par le modèle à tester, une procédure d'autotest (E8) et recevoir un mot de résultat (OK / ERROR),

- prendre une décision (E10) : si le résultat précédent est égal à la première valeur (OK) alors réaliser une étape E12, sinon, si le résultat précédent est égal à la deuxième valeur (ERROR), réaliser une étape E14,

- prendre une décision (E12) : si une autre procédure d'autotest doit être exécutée, alors réaliser une nouvelle étape E8, sinon, fin du procédé, et

- mémoriser (E14) des informations relatives à la procédure d'autotest précédemment effectuée, les informations contenant notamment une adresse à laquelle une erreur a été détectée, puis fin du procédé.

5. Procédé selon la revendication 4, caractérisé en ce qu'il comprend en outre les étapes d'initialisation suivantes, à exécuter avant la première étape E8 :

- recevoir des spécifications d'un utilisateur (E'2) concernant au moins deux procédures d'autotest,

- lire (E'4), dans une table, des données caractéristiques du modèle de processeur à tester nécessaires à l'exécution des procédures d'autotest,

- exécuter (E'6) plusieurs étapes E6, de manière successive, chaque étape E6 correspondant à une procédure d'autotest.

5        6. Procédé selon l'une des revendications 4 ou 5, caractérisé en ce qu'il comprend en outre l'étape E7 suivante, exécutée après l'étape E'6, et consistant à :

10        - fournir une étude statistique (E7) des instructions à tester lors des étapes E8 suivantes, pour estimer la couverture du jeu d'instructions et des performances du modèle de processeur à valider.

15        7. Procédé selon l'une des revendications 4 à 6, caractérisé en ce qu'il comprend en outre l'étape E16 suivante, effectuée à la fin du procédé et consistant à :

20        - fournir une étude statistique (E16) des résultats (OK / ERROR) fournis lors de toutes les procédures d'autotests (E8) exécutées par le modèle à tester.

25        8. Procédé de génération de programmes d'autotest, caractérisé en ce qu'il comprend les étapes E2 à E6 du procédé selon l'une des revendications 1 à 7, et en ce qu'il comprend en outre l'étape E'8 suivante :

30        - écrire (E'8) un programme d'autotest pour faire exécuter une procédure d'autotest (E8) par le modèle de processeur.

30        9. Procédé selon la revendication 8, caractérisé en ce que lors de la réalisation de l'étape E'8, le programme d'autotest est écrit dans un langage de type assembleur, compréhensible et exécutable par tous les modèles à tester d'un même processeur.

° 10. Procédé selon l'une des revendications 8 ou 9, caractérisé en ce que l'étape E81 est réalisée à partir d'instructions du jeu d'instructions du modèle à tester.

- 5        ° 11. Procédé selon l'une des revendications 8 à 10, caractérisé en ce qu'il est mis en œuvre sous la forme d'un programme écrit dans un langage évolué de type DGL et/ou C++, compréhensible par l'utilisateur.

